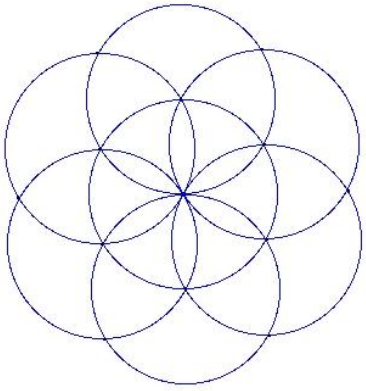
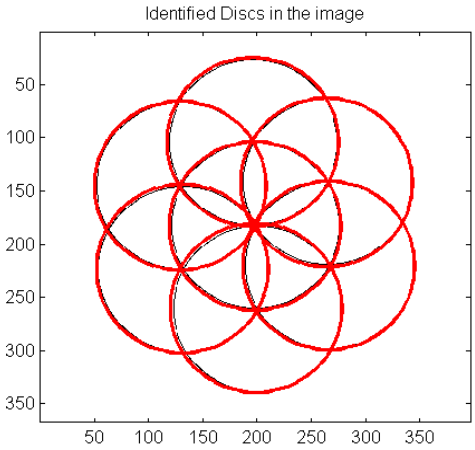
Report for Assignment 1: Hough Transform
Ravikiran Janardhana
Medical Image Analysis – COMP 775
Fall 2011

Tasks completed

1. Implemented Hough transform for circles with the required parameters. Below is an example image where Hough transform has been performed and the resultant circles are superimposed on the original image.



Original Image



Discs detected after Hough Transform (r=80px, σ(Gaussian) = σ(Parzen) = 2.1, grad_mag_threshold=0.2)

2. Implemented the program to generate test images with randomly positioned discs given discrete disk radius, a list of possible intensities, number of discs, blurring level and intensity noise levels. Below is the example images produced where intensities were chosen randomly from a given set.

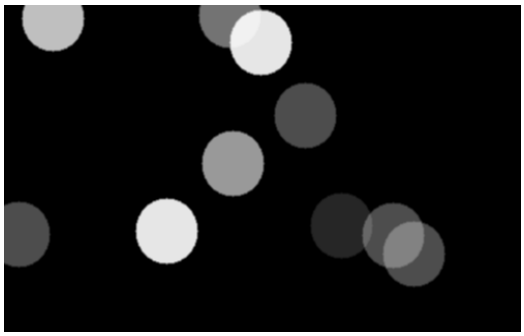
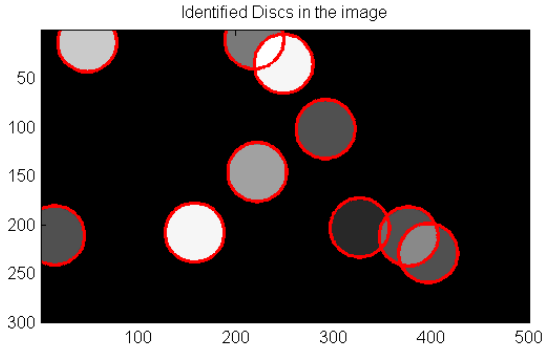


Image with blur and no noise



Discs detected after hough transform (r=20px, σ(Gaussian)=σ(Parzen) = 3.1, grad_mag_threshold=0.35)

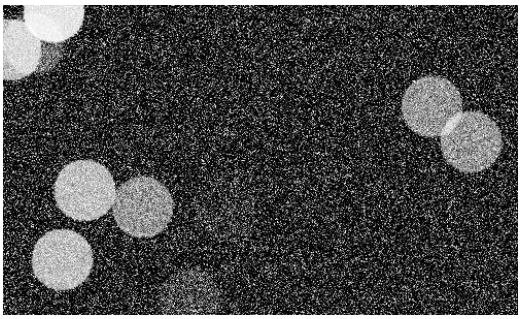
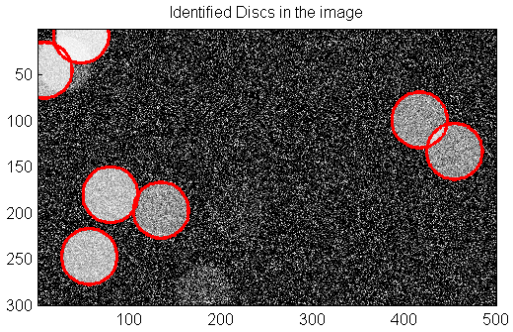


Image with blur and noise=0.2



Discs detected after Hough transform (r=20px, σ(Gaussian) = σ(Parzen) = 3.4, grad_mag_threshold=0.3)

Comments

- In order to detect discs efficiently amidst blur and noise, the Gaussian standard deviation (2 <= σ <= 4), Parzen standard deviation (2 <= σ <= 4) and the gradient magnitude threshold (0.25 <= grad_mag_threshold <= 0.4) parameters had to be increased suitably to reduce the effect the noise.
- The overlapped discs were detected efficiently as seen in the above figure.
- However, when the noise level was increased to a value greater than 0.2, the circle detection went awry and smoothing the image further didn't yield any great result.

3. Downloaded images from the web and applied Hough transform on them to locate discs/circles.

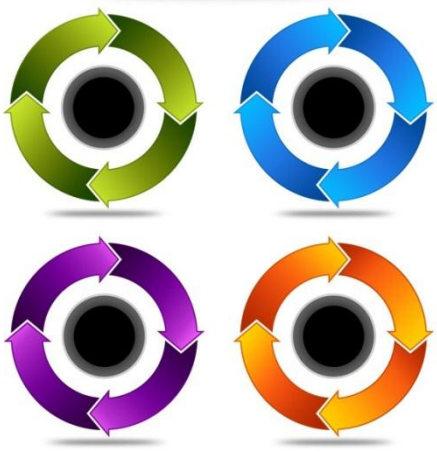
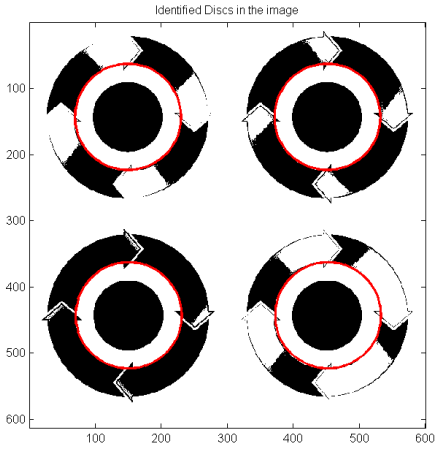


Image from the internet



Discs detected after Hough transform (r=80px, σ(Gaussian)=σ(Parzen)=2.1, gradmag_threshold=0.2)

Comments

- The program worked very well when the circles were clearly distinguishable.
- The input radius had to be within +10/-10 pixels of the actual radius of the circles in the image for the program to work efficiently.
- However, in images where there was noise, I had to increase the Gaussian standard deviation ($2 <= \sigma <= 4$), Gradient Magnitude Threshold ($0.25 <=$ grad_mag_threshold $<= 0.4$) and Parzen standard deviation ($2 <= \sigma <= 4$) to smoothen it more so that the circles would be located accurately.

Issues faced

1. Took me sometime to figure out how to compute the center of a circle given the polarity and the gradient magnitude/direction.
2. Found it difficult to implement the sigmoid function and hence used the already available "sigmf" function in Matlab to achieve the same.

Code

1. hough_circle_demo.m – The main program which is used to demo the Hough transform. Initially, the program asks the user whether he wants to choose an existing image or generate an image. Depending on the choice, the user further chooses parameters to perform Hough transform and the result is displayed. The user can continue to detect more circles after the first result is displayed or quit if satisfied.
2. hough_transform_circle.m – The heart of the code is in this file which performs Hough transform and displays the result.
3. generate_disk_image.m – This program is used to generate discs based on radius, intensities, no. of discs, blur level and noise levels.

Algorithm

1. Apply a Gaussian convolution
2. Find the gradient (FX, FY) along X and Y directions
3. Find the Gradient Magnitude [sqrt(FX^2 + FY^2)].

4. For each pixel, check if it is edgy enough (gradient magnitude > gradient magnitude threshold), if yes, find the gradient direction and compute the center of the circle based on the polarity value.
5. Find the vote strength of the pixel using sigmoid function and update the accumulator.
6. Blur the accumulator with Parzen standard deviation parameter (Parzen Windowing).
7. Find the maximum intensity in the blurred accumulator and find the center position and superimpose the circle at this point with input radius on the original image.
8. Reset the value of center position and its surrounding pixels in the accumulator to 0.0 to avoid finding another maximum just near the previous maximum.
9. Ask if the user wants to find another circle with the new accumulator, if yes go to Step 6, else go to Step 10.
10. Stop.

Some more results

Example 1

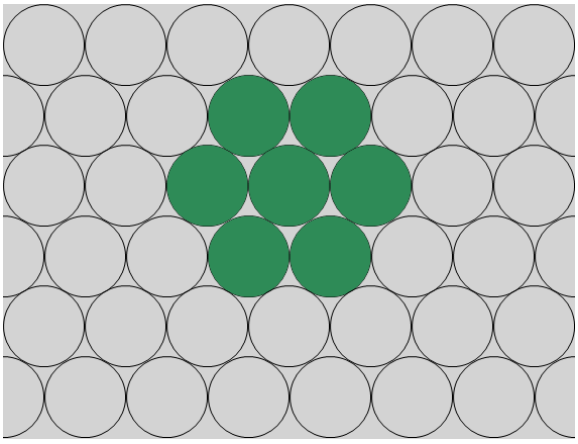
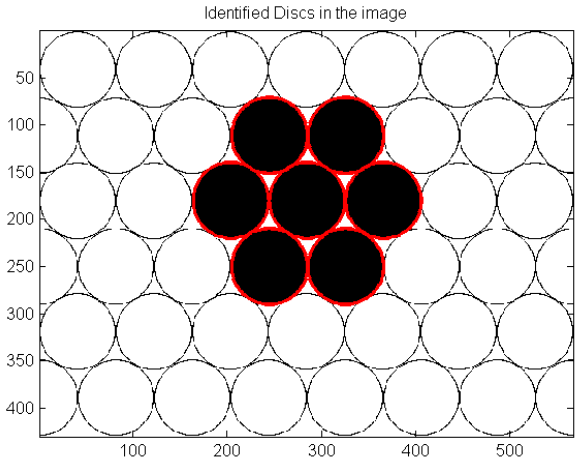


Image from Internet



Discs detected (dark on white) after Hough Transform (r =40, σ(Gaussian)=σ(Parzen)=2.1, gradmag_threshold=0.2)

Example 2

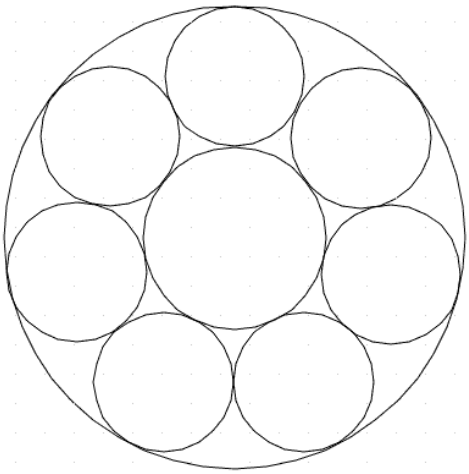
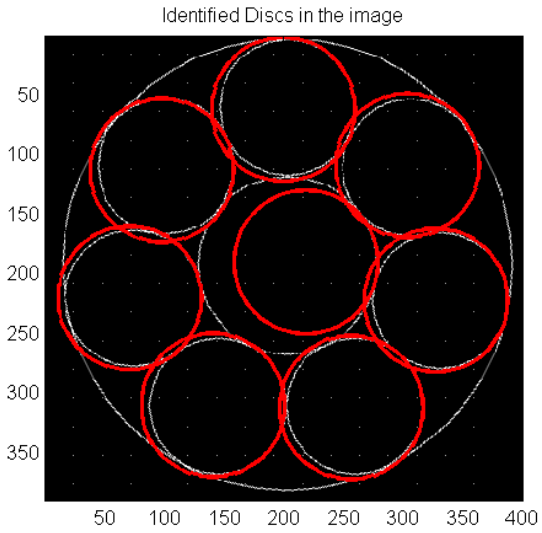


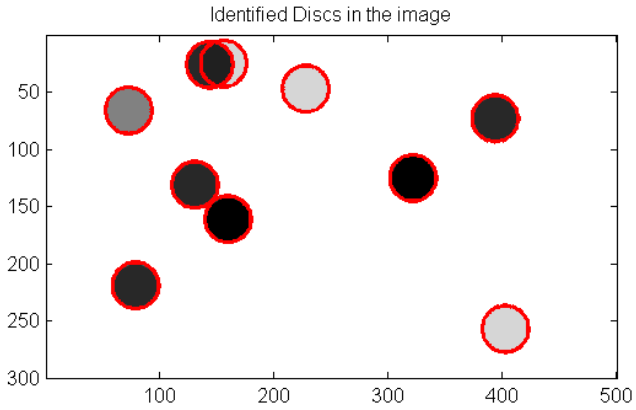
Image from the internet



Discs detected after Hough Transform
(r =60, σ(Gaussian)=σ(Parzen)=2.1,
gradmag_threshold=0.2)

Example 3

Generated image with blur



Discs detected after Hough Transform (r=20)
(r =20, σ(Gaussian)=σ(Parzen)=3.4,
gradmag_threshold=0.35)